Today: switched Ethernet

- not shared bus anymore

  → every device connected by point-to-point link to switch

  → switch: a computer

  → with special hardware support to increase packet processing speed

- arriving Ethernet frames may be scheduled

  → buffering: who goes first (e.g., FIFO, priority)

- no more physical collision

  → frames don't collide anymore

  → frames arrive on separate point-to-point links connected to separate NICs: called ports on switches

  → instead of collision: buffer overflow

Note: a switch has nothing to do with CSMA/CD

→ CS, MA, CD are protocols designed for shared buses

→ but CSMA/CD still used in switched Ethernet

→ why?

Real-world importance of backward compatibility:

- legacy Ethernet NICs speak CSMA/CD
- switched Ethernet introduced in the early-to-mid 1990s must interoperate with legacy NICs
- otherwise can't sell in the real world

  → critical constraint of new networking technologies

  → not just networking though

How to achieve backward compatibility:

→ Ethernet switch emulates CSMA/CD

- emulate CSMA/CD protocol—even though there is no bus sharing and collision
- goal: legacy CSMA/CD NIC cannot tell difference

  → switch or bus

  → as if connected to a bus

- meaning of collision: upon buffer overflow, send collision signal

  → jam signal

- legacy NIC responds with retransmit and exponential backoff

End result: can connect many devices without suffering heavy collision of shared buses

Long distance Ethernet: e.g., 1000Base-LX

$\longrightarrow$ what about length limit of CSMA/CD?

Medium-haul GigE/10GigE (802.3ae): 500m, 5km, 40km

- solution: disable CSMA/CD

  $\rightarrow$ switch-to-switch: disable at both ends

  $\rightarrow$ purely point-to-point link

  $\rightarrow$ backward compatibility: not an issue anymore

- flow control

  $\rightarrow$ send pause frame to prevent buffer overflow

- called carrier or long-haul Ethernet

Today: Ethernet is both local-area and metropolitan/wide-area network technology

FDDI (Fiber Distributed Data Interface)

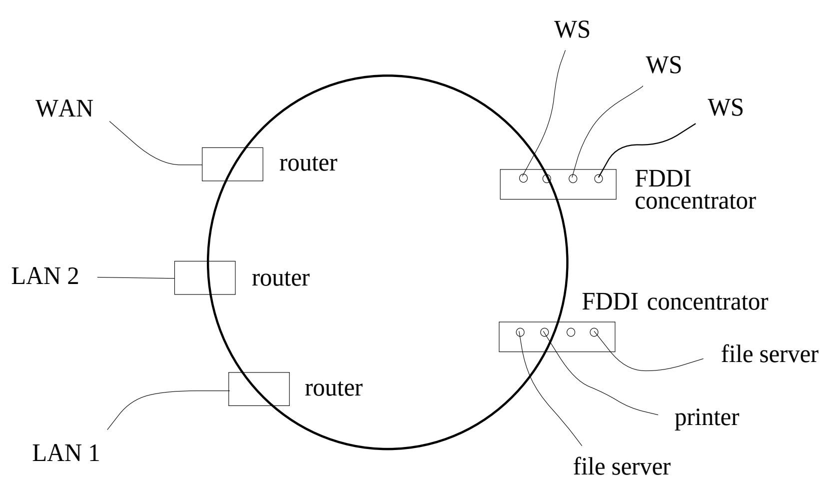
⟶ token ring architecture

High-bandwidth extension of IBM 4 Mbps and 16 Mbps IEEE 802.5 token ring standard.

⟶ 100 Mbps bandwidth

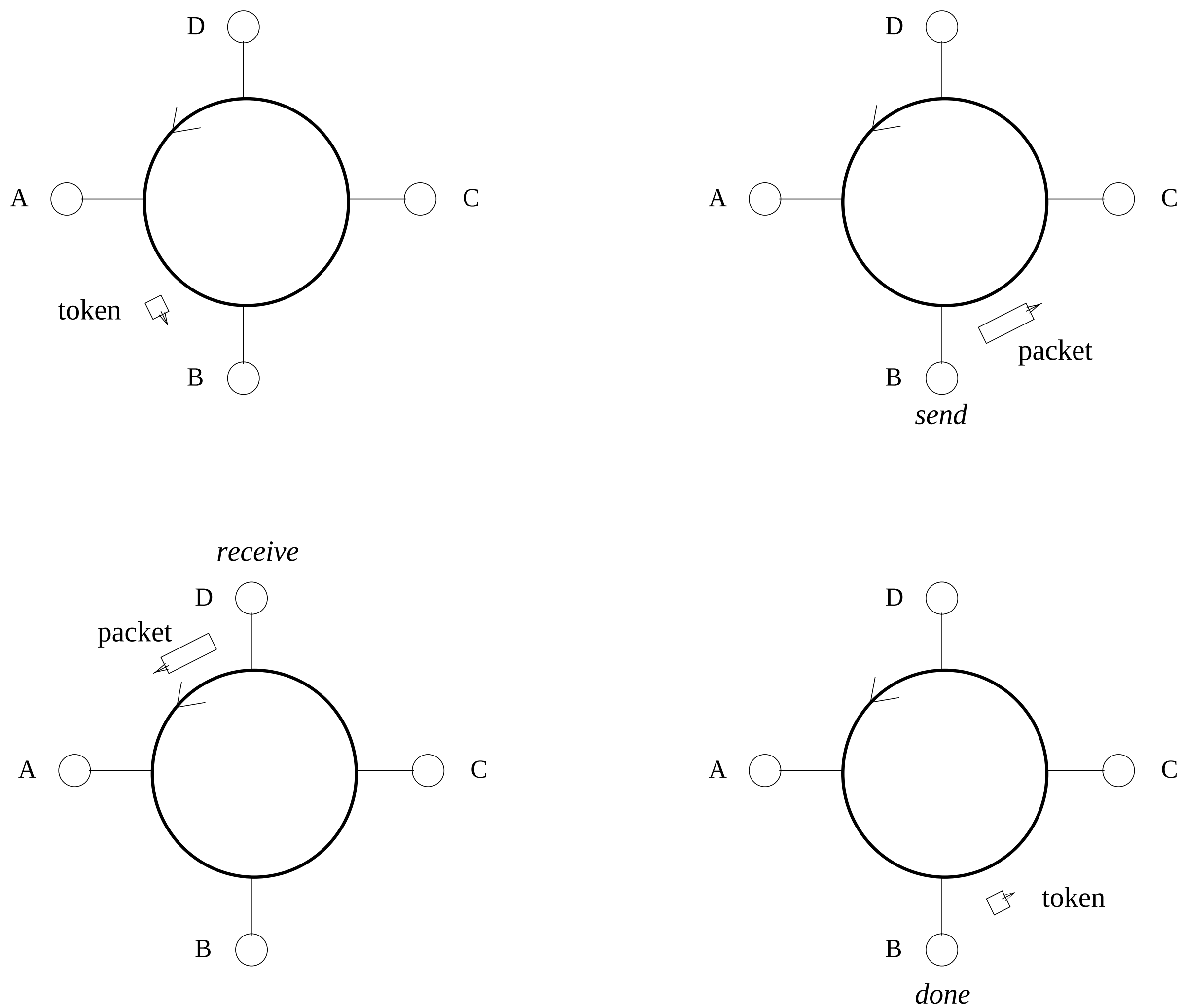
Used as high-bandwidth campus/city backbone.

⟶ metropolitan/campus distance: MAN

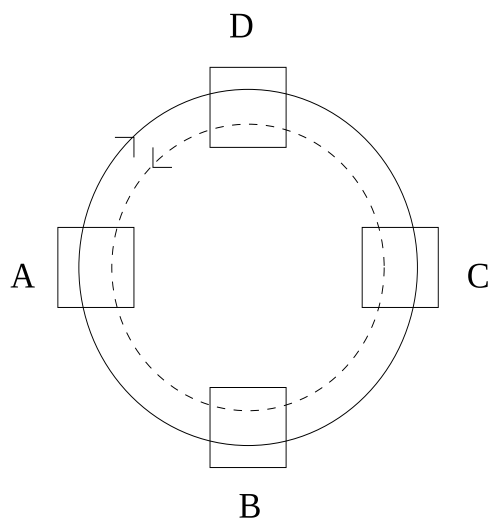


Basic operation:

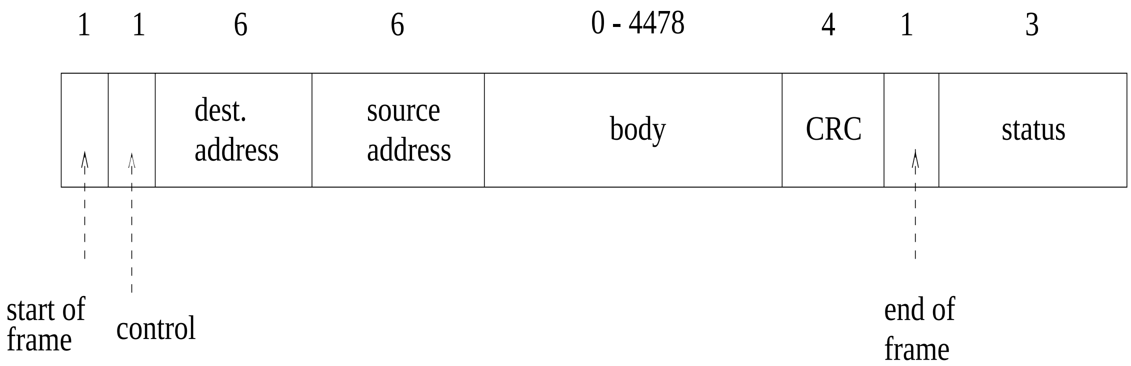
$\longrightarrow$ B wants to send to D



Fault-tolerance:



- DAS (dual attachment station)
- SAS (single attachment station)



- frame size $<$ 4500 B
- 4B/5B encoding
- synchronous/asynchronous data
- 2 km inter-station distance
- 200 km diameter (multimode fiber); 100 km circumference

Performance issues: fairness and efficiency

- TRT (token rotation time)
- THT (token holding time)

$$\text{TRT} = \text{no. of nodes} \times (\text{THT} + \text{link latency})$$

To increase efficiency: increase THT

$\longrightarrow$ let station send as much as it needs

$\longrightarrow$ same as frame size $\uparrow$

$\longrightarrow$ THT $\uparrow \implies$ utilization $\rho \uparrow$

To increase fairness: limit THT

$\longrightarrow$ limit station's one-time sending of data

To facilitate fairness: introduce TTRT (target token rotation time).

THT determining factor:

- prioritized frames: synchronous/asynchronous

  $\rightarrow$ synchronous: voice
- synchronous frames always get sent
- if TRT $>$ TTRT, then late; don't send asynchronous data
- if TRT $\leq$ TTRT, then early; send asynchronous data

How to set TTRT?

⟶ token claim process

⟶ initiate when needed (e.g., start-up)

- each station submits claim frame containing TTRT bid
- smaller TTRT bid overrides higher TTRT bids
- winner: smallest bid value when claim frame has made full circle

  → called leader election

At the end of the day, consistent TTRT value among all stations.

→ called consensus problem

Compare against Ethernet's CSMA/CD

- $\longrightarrow$ round-robin reservation
- $\longrightarrow$ absence of MA and collision
- $\longrightarrow$ determinism vs. indeterminism
- $\longrightarrow$ still: imperfect QoS assurance